Applicant : Blumenthal et al.
Serial No. : 10/659,056
Filed : September 9, 2003
Page : 3 of 14

Attorney's Docket No.: 13913-083001 / 2002P10217
US

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A computer program product, tangibly embodied on~~an information carrier~~ a machine-readable storage device, comprising instructions operable to cause data processing apparatus to:

establish ~~any number~~ a plurality of checkpoints in a first computer program, the first computer program having a program structure; and

~~include~~ assign each checkpoint in the plurality of checkpoints to a checkpoint group without regard to the program structure of the first computer program, ~~wherein each checkpoint group can include any number of the checkpoints regardless of where the checkpoints are in the first computer program~~ wherein the assignment of a given checkpoint to a respective checkpoint group can be made independently of the location of the checkpoint, so that the structure of checkpoint groups is independent of the program structure of the first computer program.

2. (Original) The product of claim 1, wherein the checkpoints comprise assertion statements and breakpoint statements.

3. (Currently amended) The product of claim 1, further comprising instructions to:

establish activation variants to enable multiple checkpoint groups ~~or compilation units or both~~ to be managed jointly.

4. (Original) The product of claim 1, further comprising instructions to:

receive a control input activating a first checkpoint group; and

activate the checkpoints in the first checkpoint group.

5. (Currently amended) The product of claim 4, wherein the instructions to receive a control input further comprise instructions to:

receive a control input that specifies a mode ~~and the mode comprises one of:~~ in which ~~activating~~ checkpoints that are assertions ~~to~~ terminate on assertion failure;

receive a control input that specifies a mode in which ~~activating~~ checkpoints that are assertions ~~to~~ log status on assertion failure; and

receive a control input that specifies a mode of activating checkpoints ~~that are~~ in which assertions ~~to~~ break in a debugger on assertion failure.

6. (Currently amended) The product of claim 4, further comprise instructions to:

receive a control input specifying ~~a scope~~ that activating is to be performed only for a particular user of the first computer program.

7. (Currently amended) The product of claim 4, further comprise instructions to:

receive a control input specifying ~~a scope specifying that activating is to be performed only for a particular user of the first computer program, and~~ that activating is to be performed only for a particular server on which the first computer program is running~~, or that activating is to be performed globally~~.

8. (Original) The product of claim 1, wherein the checkpoints and the first computer program are in a source code form.

9. (Original) The product of claim 8, wherein:

the checkpoints comprise assertion statements, each assertion statement when activated testing whether a specified assertion condition is true or false; and

the checkpoints comprise breakpoint statements, each breakpoint statement when activated halting program execution when it is encountered during program execution.

10. (Original) The product of claim 8, wherein:

the assertion statements comprise an assertion statement having an argument to activate logging with programmer-controlled granularity.

Applicant : Blumenthal et al.
Serial No. : 10/659,056
Filed : September 9, 2003
Page : 5 of 14

Attorney's Docket No.: 13913-083001 / 2002P10217
US

11. (Original) The product of claim 8, further comprising instructions to establish a development environment for developing the first computer program in which the checkpoint groups are development objects.

12. (Original) The product of claim 1, wherein the checkpoints and the first computer program are in a compiled form.

13. (Currently amended) Apparatus, comprising:

means for establishing ~~any number~~ a plurality of checkpoints in a computer program, the computer program having a program structure; and

means for ~~including~~ assigning each checkpoint in the plurality of checkpoints to a checkpoint group without regard to the program structure of the computer program, wherein ~~each checkpoint group can include any number of the checkpoints regardless of where the checkpoints are in the computer program~~ the assignment of a given checkpoint to a respective checkpoint group can be made independently of the location of the checkpoint, so that the structure of checkpoint groups can be independent of the program structure of the first computer program.

14. (Currently amended) The apparatus of claim ~~12~~ 13, wherein:

the checkpoints comprise assertions and breakpoints.

15. (Currently amended) The apparatus of claim ~~12~~ 13, further comprising:

means for associating an activation variant with a checkpoint group.

16. (Currently amended) The apparatus of claim ~~12~~ 13, further comprising:

means for associating an activation variant with a compilation unit.

17. (Currently amended) A method, comprising:

    receiving a computer program having a plurality of checkpoints assigned to a plurality of checkpoint groups, each checkpoint and each checkpoint group being identified by a group identifier, the computer program having a program structure, each group identifier identifying checkpoints without limitation as to the location of the checkpoints in the computer program, so that the structure of checkpoint groups is independent of the program structure of the computer program, each checkpoint being an assertion or a breakpoint; and

    receiving user input to invoke checkpoints as a group according to their group identifiers.

18. (Currently amended) The method of claim 16 17, further comprising:

    receiving a user input specifying a mode of invocation of checkpoints; and

    invoking checkpoints according to the specified mode.

19. (Currently amended) The method of claim 16 17, further comprising:

    receiving a further user input specifying a scope of invocation of checkpoints, the scope specifying that checkpoints are to be invoked only for a particular user of the first computer program, that checkpoints are to be invoked only for a particular server on which the first computer program is running, or that checkpoints are to be invoked globally; and

    invoking checkpoints according to the specified scope.

20. (New) The method of claim 17, further comprising:

    receiving a further user input specifying a scope of invocation of checkpoints, the scope specifying that checkpoints are to be invoked only for a particular server on which the computer program is running; and

    invoking checkpoints according to the specified scope.

21. (New) The product of claim 1, wherein:

    the instructions to include each checkpoint in a checkpoint group comprise instructions to associate a respective checkpoint group name with each checkpoint in the source code for the first computer program.

22. (New) The apparatus of claim 13, wherein:

    the means for including each checkpoint in a checkpoint group comprises means for associating a respective checkpoint group name with each checkpoint in the source code for the computer program.

23. (New) The method of claim 17, wherein:

    the checkpoints identified by a group identifier are each identified by the respective group identifier in the source code for the computer program.

24. (New) The method of claim 17, wherein the computer program has checkpoints including both assertions and breakpoints.

25. (New) A method for adding checkpoints to a computer program having source code, the method comprising:

    adding to the computer program a plurality of checkpoints each assigned to a checkpoint group by a respective group name for the checkpoint, the assignment of checkpoints to checkpoint groups being made without regard to the program structure of the computer program.

26. (New) The method of claim 25, further comprising:

    adding the plurality of checkpoints to the source code of the computer program, the respective group name for each checkpoint being included in the source code for the checkpoint; and

    transporting the checkpoint groups as development objects with the computer program from a development environment to a production environment, the development objects being objects created and managed by the development environment.

27. (New) The product of claim 10, wherein:

    the argument to activate logging indicates that a log entry is made for each distinct value of a named field.